CLAIMS

SUB A17

1. A system for synchronizing a visualization with audio samples comprising:

one or more audio sources configured to provide audio samples that are to be rendered by a media player;

an audio sample pre-processor communicatively linked with the one or more audio sources and configured to receive and pre-process audio samples before the samples are rendered, the pre-processing providing characterizing data associated with each sample; and

one or more effects configured to receive the characterizing data and use the characterizing data to render a visualization that is synchronized with an audio sample that is being rendered by the media player.

2. The system of claim 1 further comprising multiple data structures configured to hold the characterizing data, each data structure being associated with an audio sample.

3. The system of claim 2, wherein the audio sample pre-processor is configured to maintain the data structures.

4. The system of claim 2, wherein the audio sample pre-processor comprises a timestamp module that provides a timestamp for each audio sample, each timestamp being maintained by a data structure associated with the audio sample.

5. The system of claim 4, wherein the timestamp is assigned by the timestamp module based upon when the audio sample is calculated to be rendered by the media player.

6. The system of claim 4, wherein the audio sample pre-processor is configured to:

query a media player audio sample renderer for a time associated with an audio sample that is being currently rendered, and

use the time to ascertain a timestamp of an associated audio sample, the audio sample pre-processor further being configured to provide characterizing data of the associated audio sample so that the characterizing data can be used to render the visualization.

7. The system of claim 1, wherein said characterizing data comprises frequency data.

8. The system of claim 1, wherein said audio sample pre-processor comprises a Fast Fourier Transform that it utilizes to process the audio samples to provide frequency data associated with the audio samples.

9. A media player comprising:

an audio sample pre-processor configured to receive and pre-process audio samples before the samples are rendered by the media player, the pre-processing providing frequency data associated with each sample; and one or more effects configured to receive the frequency data and use the frequency data to render a visualization that is synchronized with an audio sample that is being rendered by the media player.

10. The media player of claim 9 further comprising multiple data structures configured to hold the frequency data, each data structure being associated with an audio sample.

11. The media player of claim 10, wherein the audio sample pre-processor comprises a timestamp module that provides a timestamp for each audio sample, each timestamp being maintained by a data structure associated with the audio sample, and further wherein the audio sample pre-processor is configured to:

query a media player audio sample renderer for a time associated with an audio sample that is being currently rendered, and

use the time to ascertain a timestamp of an associated audio sample, the audio sample pre-processor further being configured to provide frequency data of the associated audio sample to the one or more effects so that the frequency data can be used to render the visualization.

12. The media player of claim 9, wherein the audio sample pre-processor pre-processes the audio samples by using a Fast Fourier Transform to provide the frequency data.

~~**13.** A system for synchronizing a visualization with audio samples comprising:~~

~~an audio sample pre-processor configured to receive and preprocess audio samples before the samples are rendered by a renderer that comprises part of a media player, the audio sample preprocessor preprocessing the samples to provide characterizing data associated with each sample, the characterizing data comprising a timestamp associated with each audio sample, the timestamp being assigned in accordance with when the audio sample is calculated to be rendered by the renderer;~~

~~multiple data structures configured to hold the characterizing data, each data structure being associated with an audio sample;~~

~~an audio rendering object configured to call the audio sample pre-processor to ascertain the characterizing data associated with an audio sample that is currently being rendered by the renderer;~~

~~the audio sample pre-processor being configured to ascertain said characterizing data by querying the renderer for a time associated with the currently-rendered audio sample, and then using said time to identify a data structure having a timestamp that is nearest in value to said time; and~~

~~one or more effects configured to receive characterizing data that is associated with the data structure having the timestamp that is nearest in value to said time, and use the characterizing data to render a visualization that is synchronized with the audio sample that is being rendered by the renderer.~~

~~**14.** The system of claim 13, wherein the characterizing data comprises frequency data.~~

15. The system of claim 13, wherein the audio sample pre-processor comprises a Fast Fourier Transform that it utilizes to process the audio samples to provide frequency data associated with the audio samples.

16. The system of claim 13, wherein the visualization is rendered in a rendering area in which other media types can be rendered.

17. The system of claim 16, wherein the other media types comprise a video type.

18. The system of claim 16, wherein the other media types comprise a skin type.

19. The system of claim 16, wherein the other media types comprise a HTML type.

20. The system of claim 16, wherein the other media types comprise a animation type.

21. A system for processing audio samples comprising:

a timestamp module for assigning timestamps to audio samples that are to be rendered by a media player renderer;

a spectrum analyzer for processing the audio samples to provide frequency data associated with the audio samples;

multiple data structures each of which being associated with an audio sample, the data structures each containing timestamp data and frequency data for its associated audio sample; and

the system being configured to use the timestamp data to ascertain a data structure associated with an audio sample that is currently being rendered by the media player renderer and provide the frequency data associated with that audio sample so that the frequency data can be used to render a visualization associated with that audio sample.

22. The system of claim 21, wherein the spectrum analyzer comprises a Fast Fourier Transform that is utilized to provide the frequency data.

23. A method of providing a visualization comprising:

receiving multiple audio samples;

pre-processing the audio samples before they are rendered by a media player renderer, the pre-processing providing characterizing data for each sample;

determining when an audio sample is being rendered by the media player renderer; and

responsive to said determining, using the characterizing data that is associated with the audio sample that is being rendered to provide a visualization.

24. The method of claim 23 further comprising maintaining characterizing data for each audio sample in a data structure associated with each audio sample.

09871405.062501

25. The method of claim 24, wherein the characterizing data comprises a timestamp associated with the audio sample, the timestamp being provided based upon when the audio sample is calculated to be rendered by the media player renderer.

26. The method of claim 25, wherein said determining comprises:

ascertaining a time associated with a currently-rendered audio sample;

selecting a data structure having a timestamp that is nearest the time; and

providing characterizing data associated with the selected data structure to a component configured to provide the visualization.

27. The method of claim 23, wherein the characterizing data comprises frequency data associated with each sample.

28. The method of claim 23, wherein said pre-processing comprises using a Fast Fourier Transform to provide frequency data associated with the samples.

29. A method of providing a visualization comprising:

receiving multiple audio samples;

pre-processing the audio samples before they are rendered by a media player renderer, the pre-processing comprising at least (1) using a Fast Fourier Transform to provide frequency data associated with the samples, and (2) associating a timestamp with each sample;

maintaining frequency data and a timestamp for each sample in a data structure;

determining when an audio sample is being rendered by a media player renderer by:

ascertaining a time associated with a currently-rendered sample; and

selecting a data structure having a timestamp that is nearest the time; and

providing characterizing data associated with the selected data structure to a component configured to provide the visualization.

30. One or more computer-readable media having computer-readable instructions thereon which, when executed by a computer, cause the computer to implement the method of claim 29.

31. A method of providing a visualization comprising:

calling an audio sample pre-processor for characterizing data that is associated with an audio sample that is currently being rendered by a media player renderer;

calling the media player renderer for a time associated with a currently-rendered audio sample;

using the time to select a data structure containing characterizing data associated with the currently-rendered audio sample; and

providing the characterizing data to a component for rendering a visualization.

~~**32.** The method of claim 31, wherein the characterizing data comprises frequency data associated with the audio samples.~~

~~**33.** The method of claim 31, wherein the characterizing data comprises frequency data associated with the audio samples and generated by pre-processing the audio samples using a Fast Fourier Transform.~~

~~**34.** One or more computer-readable media having computer-readable instructions thereon which, when executed by a computer, cause the computer to:~~

~~pre-process audio samples using a Fast Fourier Transform to provide frequency data, the audio samples being pre-processed before they are rendered by a media player renderer;~~

~~query for frequency data that is associated with an audio sample that is currently being rendered by the media player renderer;~~

~~query for a time associated with the currently-rendered audio sample;~~

~~use the time to select a data structure containing frequency data associated with the currently-rendered audio sample; and~~

~~provide the frequency data to a component for rendering a visualization.~~

~~**35.** A method of providing a visualization comprising:~~

~~defining a frame rate at which visualization frames are to be rendered;~~

~~setting a threshold associated with an amount of time that is to be spent rendering a visualization frame;~~

~~monitoring the time associated with rendering individual visualization frames;~~

determining whether a visualization frame rendering time exceeds the threshold; and

providing an effective frame rate for rendering visualization frames that is longer than the defined frame rate if the determined visualization frame rendering time exceeds the threshold.

36. The method of claim 35, wherein said providing comprises increasing a call interval associated with calls that are made to a visualization-rendering component.

37. The method of claim 35 further comprising modifying the effective frame rate so that the visualization frames are rendered at the defined frame rate.

38. One or more computer-readable media having computer-readable instructions thereon which, when executed by a computer, cause the computer to:

set a threshold associated with an amount of time that is to be spent rendering a visualization frame for a given frame rate;

monitor the time associated with rendering individual visualization frames;

determine whether a visualization frame rendering time exceeds the threshold; and

provide an effective frame rate for rendering the visualization that is longer than the defined frame rate if the determined visualization frame rendering time exceeds the threshold.